Change Point Detection and Meta-Bandits for Online Learning in Dynamic Environments

Cédric Hartland, Nicolas Baskiotis, Sylvain Gelly, Michèle Sebag, Olivier Teytaud

Change Point Detection and Meta-Bandits for Online Learning in Dynamic Environments

Cédric Hartland[1,2], Nicolas Baskiotis[1,2], Sylvain Gelly[1,2], Olivier Teytaud[1,2], Michèle Sebag[1]

[1] LRI; Univ. Paris-Sud, CNRS; F-91405 Orsay, France

[2] INRIA Futurs; projet TAO, Bât. 490, F-91405 Orsay, France
{hartland,nbaskiot,gelly,,teytaud,sebag}@lri.fr

Abstract : Motivated by realtime website optimization, this paper is about online learning in abruptly changing environments. Two extensions of the UCBT algorithm are combined in order to handle dynamic multi-armed bandits, and specifically to cope with fast variations in the rewards. Firstly, a change point detection test based on Page-Hinkley statistics is used to overcome the limitations due to the UCBT inertia. Secondly, a controlled forgetting strategy dubbed Meta-Bandit is proposed to take care of the Exploration vs Exploitation trade-off when the PH test is triggered. Extensive empirical validation shows significant improvements compared to the baseline algorithms. The paper also investigates the sensitivity of the proposed algorithm with respect to the number of available options.

1 Introduction

The Game Theory perspective is gradually becoming more relevant and appealing to Machine Learning (ML) for several reasons (Cesa-Bianchi & Lugosi, 2006). On the one hand, the size of the dataset might forbid the use of standard algorithms, calling for incremental, anytime or streaming algorithms (Cormode & Muthukrishnan, 2005). Likewise, the dynamics of the data generating process might require new learning algorithms, able to estimate on the fly the relevance of the training examples, and accommodate these relevance estimates within the learning process (Kifer *et al.*, 2004). On the other hand, the quality of the learning algorithm might be measured based on its cumulated performance as opposed to its asymptotic performance (Auer & Ortner, 2007); specifically in the context of lifelong learning, statistical analysis might focus on the regret, cumulated loss compared to the best possible behaviour, as opposed to the generalization error, measured after the end of the training phase.

This paper is motivated by realtime website optimization; the goal is to provide a community of users with the news they are most interested in. Standard recommendation systems focus on the user modelling and collaborative filtering (Grcar *et al.*, 2005).

The topic addressed in this paper is somewhat different as we focus on the dynamic changes in every user's interests. The point is not to model the environment and the hidden causes for theses changes. The user is instead formalized as a multi-armed bandit, associating a reward (his interest), to every news presented by the web site [1](Auer *et al.*, 2002) to focus the study on the best strategy to match not only the changing users interests but any types of changes.This problem has been formalized as for the Pascal's Exploration vs Exploitation challenge proposed by Touch Clarity..

Formally, a news item is viewed as an arm, the associated reward being the number of times the visitors click on it. As the goal is to maximize the total number of clicks, the website administration must achieve some trade-off between exploration (serving all news in order to identify the most popular ones) and exploitation (serving the most popular news identified so far). One extra difficulty of website optimization compared to multi-armed bandits is that the set of news and the user's interests change often and abruptly; the news on the front page should obviously depend on the actuality (e.g. elections, sport events); the users also undergo fast variations e.g. on week days or during holidays.

This paper is about online learning in dynamic environments. Though online algorithms offer some leeway for accommodating dynamic environments, empirical evidence shows that the Exploration *versus* Exploitation trade-off achieved by e.g. the UCBT algorithm (Auer *et al.*, 2002) is not appropriate for abruptly changing environments because of its inertia; UCBT is designed for stationary environments. In order to adapt online learning to such abrupt changes, two interdependent issues must be addressed. The first issue, referred to as change-point detection (Page, 1954), is to decide whether some change has occurred beyond the "natural" variations of the environment. The second issue is to design a good strategy for such change points. On the one hand, the change-point detection must trigger some extra exploration; this extra exploration relates to the (partial) forgetting of the recent history. On the other hand, if the change-point detection was a false alarm, the process should quickly recover its memory and switch back to exploitation; otherwise, the extra exploration results in wasting time.

The *Adapt-EvE* algorithm presented in this paper extends the UCBT algorithm (Auer *et al.*, 2002) with two main contributions. Firstly, *Adapt-EvE* incorporates a change-point detection test based on the Page-Hinkley statistics; parametrized after the desired false alarm rate, this test provably minimizes the expected time before detection (section 2.3). Secondly, the PH test triggers a specific transient Exploration vs Exploitation (EvE) strategy implemented as a *Meta-Bandit*. More precisely, the transient EvE is viewed as another bandit problem, where the two options are: i/ restarting UCBT from scratch ii/ discarding the change detection and keeping the same UCBT strategy as before (section 3). Empirical validation conducted on the EvE Challenge proposed by (Hussain *et al.*, 2006) demonstrates significant improvements over the baseline UCBT (Section 5); additionally, the scalability and robustness of *Adapt-EvE* w.r.t. the number of options is studied. The paper concludes with some perspectives for further research.

[1]This formalization was jointly defined by L. Newnham, Z. Hussain, P. Auer, N. Cesa-bianchi and J. Shawe-Taylor

2 State of the art

The multi-armed bandit problem is about maximizing the reward associated to different arms. The problem is usually formalized as a problem of maximizing reward from slot machines. Each lever provides a reward from an associated distribution. The objective is to maximize the overall collected reward by estimating the best rewarding machine without initial reward.

In order to make the paper self-contained, this section briefly introduces the multi-armed bandit problem, the UCBT algorithm (Auer *et al.*, 2002) and the Page-Hinkley statistics (Page, 1954) which will be used as change-point detection test in *Adapt-EvE*.

2.1 Background and Notations

A multi-armed bandit problem involves K arms or options. To each arm is associated its reward probability at time t noted $\mu_k,\ k = 1 \dots K$. At time t, the gambler selects some option based on the estimated rewards $\hat{\mu}_{k,t}$ and the estimation effort $n_{k,t}$ spent on every option. Originally, $n_{k,t}$ was set to the number $N_{k,t}$ of times the k-th option has been played during the first t moves; the reason why it is more convenient to consider $n_{k.t}$ as an estimation effort will become clear later on. The regret $\mathcal{L}(T)$ after T moves is the loss incurred by the gambler compared to the best possible strategy, i.e. playing the option with maximal reward μ_t^* at each move:

$$\mathcal{L}(T) = \sum_{k=1}^{K} N_{k,t} \times (\mu_t^* - \mu_k)$$

2.2 UCB1 and UCBT

The UCB1 and UCBT algorithms (Table 1) are theoretically and empirically well-established solutions of the multi-armed bandit problem in the stationary case (Auer *et al.*, 2002). Firstly, all options are played to initialize the reward estimates and estimation efforts. Thereafter, one iteratively selects the option with best estimated reward μ_t^* (exploitation), except when the upper confidence bound on the reward of some other option is greater than that of the best option; every option is thus played infinitely many times (exploration). The estimation effort $n_{k,t}$ is the number of times the k-th option is played in UCB1 and UCBT whereas a multiplicative discount factor $\rho < 1$ is used in Discounted UCBT (Kocsis & Szepesvari, 2006). The only difference between UCB1 and UCBT is that UCBT restricts the exploration strength through function $C(k,t)$, particularly so for options with small reward variance, empirically resulting in better performance (Auer *et al.*, 2002).

Under mild assumptions (rewards are independent and bounded with constant probability for every arm, arms are independent), UCB1 ensures that the loss expectation $\mathcal{L}(t)$ is bounded logarithmically with the number of moves t. Still, UCB1 and UCBT alike are not well suited to dynamic environments; the time needed before playing some (non optimal) k-th option increases with its margin $\mu_t^* - \mu_{k,t}$ and with the estimation effort $n_{k,t}$ spent on this option. In other words, UCBT algorithms need a long time to adjust the reward estimates if some change occurs after a period of stability.

UCB1(ρ)

Initialization:

$t = K$

For $k = 1 \dots K$,

Play option k, set $\hat{\mu}_{k,t}$ to the reward, $n_{k,t} = 1$

Repeat

Play k = argmax$_{j=1}^{K}$ $\hat{\mu}_{j,t} + \sqrt{\frac{2 \log (\sum_i n_{i,t}).C(j,t)}{n_{j,t}}}$

Let r be the associated reward

$n_{k,t+1} = \rho\, n_{k,t} + 1$ // effort update

$\hat{\mu}_{k,t+1} = \hat{\mu}_{k,t} + \frac{r - \hat{\mu}_{k,t}}{n_{k,t+1}}$ // reward update

For $j \neq k$, $\hat{\mu}_{j,t+1} = \hat{\mu}_{j,t}$; $n_{j,t+1} = \rho\, n_{j,t}$

$t := t + 1$

Function $C(j,t)$

if UCB1 : Return 1

if UCBT : Return $min(1/4, Var(\mu_{j,t}))$

Table 1: The UCBT algorithm skeleton. Multiplicative discount factor ρ is set to 1 for UCB1 and UCBT. The exploration strength is decreased in UCBT by using an upper confidence bound $Var(\mu_{j,t})$ on the reward variance.

Some attempts have been done to overcome UCBT inertia using discount factors ($\rho < 1$) and more generally to adapt UCBT to changing or adversarial environments (Kocsis & Szepesvari, 2006; Auer *et al.*, 1995; Kocsis & Szepesvari, 2005). However, the question of adjusting the discount factors remains. While these can indeed be optimized offline if the environment dynamics are sufficiently regular, some self-adjustment seems to be required in order to enable different exploration vs exploitation trade-offs.

Another possibility, explored in the rest of this paper, is based on the explicit detection of changes in the environment.

2.3 Change point detection

The change-point detection problem has been intensively studied in the literature, motivated by applications in meteorology, finance, video segmentation (Piriou *et al.*, 2004) or aggro-alimentary systems (Mouss *et al.*, 2004) to name a few. The studies usually incorporate some prior knowledge about the stationarity of the underlying phenomenon. In the dynamic multi-armed bandit problem, let us assume that at time t the best current option k^* is correctly identified together with the associated reward μ_t^*. There are three possible types of change. Firstly, reward μ_t^* changes although the best option remains k^*; secondly, μ_t^* abruptly decreases and another option becomes the best one; thirdly, μ_t^* does not change but the reward of some other j-th option increases to the point that j becomes the best option. Only the first two types of change will be considered in this paper, leaving the third type for further study.

Formally, let $r_1, ...r_T$ denote the rewards gathered the last T times option k^* was played. The question is whether this series can be attributed to a single statistical law (null hypothesis); otherwise (change-point detection) the series demonstrates a change in the statistical law underlying the rewards. A standard test for the above hypothesis is the Page-Hinkley (PH) statistics (Page, 1954; Hinkley, 1969, 1970, 1971; Basseville, 1988). Let $\bar{r}_t$ denote the average of $r_1, \dots r_t$ and let e_t denote the difference $r_t - \bar{r}_t + \delta$, where δ is a tolerance parameter (Piriou *et al.*, 2004). The baseline PH statistical test considers the random variable m_T defined as the sum of $e_1, \dots, e_T$. The maximum value M_T of the m_t for $t = 1 \dots T$ is also computed and the difference between M_T and m_T is monitored; when this difference is greater than a given threshold λ (depending on the desired false alarm rate), the null hypothesis is rejected i.e. the PH test concludes that a change point occurred:

$$\begin{aligned} &\bar{r}_t = \frac{1}{t}\sum_{\ell=1}^{t} r_\ell \\ &m_T = \sum_{t=1}^{T}(r_t - \bar{r}_t + \delta) \\ &M_T = \max\{m_t,\ t = 1 \dots T\} \\ &PH_T = M_T - m_T \\ &\text{Return } (PH_T > \lambda) \end{aligned} \tag{1}$$

The PH test involves two parameters. Parameter λ controls the trade-off between type I and type II errors and, equivalently, between exploration and exploitation. One strong property of the PH test is that it provably minimizes the expected time before change detection for a given false detection rate (Lorden, 1971; Moustakides, 1986; Dragalin *et al.*, 1999 part I2000 part II; Hadjiliadis & Moustakides, 2006) under reasonable assumptions. Parameter δ is meant to make the PH-test more robust when dealing with slowly varying environments. Both parameters are commonly adjusted after inspecting typical curves under the null hypothesis. Algorithmically, the PH statistics is computed recursively in a very efficient manner:

$$\begin{aligned} PH_0 &= M_0 - m_0 = 0 \\ PH_t &= M_t - m_t = \max(PH_{t-1} - r_t + \bar{r}_t - \delta, 0) \end{aligned}$$

3 Overview of *Adapt-EvE*

In order to handle abrupt changes in the environment, the *Adapt-EvE* algorithm extends the core UCBT algorithm with two modules. Firstly, the PH test is used to detect the changes in the environment. Secondly, when the change-point detection test is positive, the Exploration vs Exploitation (EvE) trade-off needs to be reconsidered. Actually, the fact that the change-point detection test is positive can be interpreted in several ways: it might be a false alarm; or it might be caused by a slow variation in the environment; or it might result from an abrupt variation in the environment.

The first two cases are addressed using some modifications in the original PH test in order to better deal with slow variations of the environment (section 3.1). In the last case, the problem is formalized as a Meta Exploration vs Exploitation (Meta-EvE) dilemma (section 3.2).

3.1 Adapting the Page-Hinkley Test

In order to avoid false alarms, one can only increase the values of the tolerance parameter δ and/or the threshold λ; the (standard) counterpart is that this increase would delay the detection of a true change.

Let us now consider the case of a slow variation in the environment. While the PH test would detect the slow increase or decrease of the best reward μ_t^*, it is clear that the core UCBT can naturally take care of such variations, and gently update μ_t^* as long as the best option remains unchanged.

These remarks suggest that the PH test should not be triggered in case of slow variations, although the test itself should not be relaxed through increasing the value of the PH parameters. The proposed solution is to decrease the inertia of the test, using a discount factor in the computation of m_t ; formally, eq (1) is replaced by:

$$m_t = \rho\, m_{t-1} + \bar{r}_t - r_t + \delta$$

3.2 Meta Exploration vs Exploitation

If the change-point detection results from an abrupt variation in the environment, some extra exploration is needed as the optimal option is likely to change; the γ-restart strategy proceeds by locally decreasing the inertia of the core UCBT, or restarting it from scratch.

On the other hand such a restart would entail some waste of time if the change-point detection was actually a false alarm. As an alternative to the γ-restart strategy (defined in next section), the dilemma between erasing or preserving the memory of the core UCBT is handled as a multi-armed bandit problem (section 3.2.2).

3.2.1 γ-Restart

The simplest way of decreasing the inertia of the UCBT algorithm is to decrease the estimation efforts for all options. Let T denote the time step when the change-point detection occurs[2]. Then, every $n_{k,T}$ is multiplied by some discount factor γ, $0 \leq \gamma < 1$; the reward estimates $\hat{\mu}_{i,t}$ are unchanged.

$$\forall k = 1 \ldots K, n_{k,T} \rightarrow \gamma\, n_{k,T}$$

Experimentally, it turns out that the optimal setting is $\gamma = 0$ (section 5); the γ-restart thus corresponds to restarting the UCBT from scratch. Experiments shown that other γ values raise the momentum, with as consequence to trigger several following change

[2]Although the PH test provides an estimate of the time step at which the distribution changes, we only considered the step T when the alarm is raised.

detections until a new best option is selected. This time consuming discount is here usually overcome by a complete restart, thus $\gamma = 0$.

3.2.2 Meta-Bandit

Another possibility is to formalize the Meta-EvE dilemma, erasing or preserving the memory of the core UCBT, as yet another multi-armed bandit problem. The first meta-option, referred to as *Old Bandit* and meant to address the false alarm case, actions the core UCBT as is (selecting the options based on the current values of $\hat{\mu}_{k,T}$ and $n_{k,T}$). The second meta-option, referred to as *New Bandit* and meant to address the true alarm case, actions a new UCBT (with $\forall\ k = 1 \ldots K, \hat{\mu}_{k,T} = n_{k,T} = 0$).

An independent UCBT referred to as Meta-Bandit is used to control the selection among *Old Bandit* and *New Bandit*. At time T, the estimation effort $n_{O,T}$ and reward $\hat{\mu}_{O,T}$ attached to *Old Bandit* (respectively, the estimation effort $n_{N,T}$ and reward $\hat{\mu}_{N,T}$ attached to *New Bandit*) are set to 0. Thereafter, Meta-Bandit decides at each time step t whether *Old Bandit* or *New Bandit* should be selected after the standard UCBT algorithm (with no discount, $\rho = 1$). The selected meta-option, say *Old Bandit* (resp. *New Bandit*), selects an option and gets some reward r accordingly (and it updates its reward estimate and estimation effort as usual). The estimate effort $n_{O,t}$ (resp. $n_{N,t}$) is incremented and the reward estimate $\hat{\mu}_{O,t}$ (resp. $\hat{\mu}_{N,t}$) is updated taking r as current reward.

Meta-Bandit thus gradually determines whether the previous change-point detection was a false alarm, through comparing the rewards of the *Old Bandit* and *New Bandit*. After MT time steps after the change-point detection occurs ($MT = 1000$ in all experiments), the best meta-option becomes the core UCBT, taking in the control of the process while the Meta-Bandit is killed. If another change occurs during this meta phase, it is not detected by the change-point detection but the low momentum of the new Bandit will allow a quick adaptation to this change in most cases.

A variant of the Meta-Bandit approach, referred as Meta-ρ-Bandit, incorporates the discount factor $\rho < 1$ within the *Old Bandit* and *New Bandit*.

4 Experimental goal and setting

This section describes our validation framework and discusses the goal of the experiments.

4.1 The EvE Challenge

As already mentioned, the extension of online algorithms to dynamic environments was motivated by a realtime website optimization application, which inspired the EvE Pascal Challenge (Hussain *et al.*, 2006). A stochastic environment simulator was devised for the purpose of this challenge, emulating the visitor behaviors and their variations; specifically, the simulator draws the probability $\mu_{k,t}(v)$ for visitor v to click on the k-th news at time t. Six types of visitors are considered independently: constant ($\mu_{k,t}(v) = C(k, v)$); frequent swap (the best option changes frequently); long Gaussian

(the best option changes after long time intervals); weekly variations ($\mu_{k,t}(v)$ varies in a coherent way, involving two sinusoidal components with different periods, the longer period being dominant and the ranking of the options changing gradually); daily variations (same as weekly variations except that the shorter period is dominant); weekly close variation (same as weekly variations plus small and short perturbations).

The algorithm proposes an option to every visitor (one for each visitor type) at every time step. For each run, the algorithm performance is its regret, computed by the environment simulator as the difference between the expected number of clicks that would have been gathered by proposing the best option for every visitor in every time step, and the number of clicks actually gathered over 10^6 time steps, representing few months. Finally, the performance reported for every algorithm and set of parameter values (see below) is the regret averaged over 100 independent runs.

The goal of experiments is to examine the algorithm robustness w.r.t. the dynamics of the environment, considering the variation of the regret over all types of visitors. Additionally, the robustness of the algorithm w.r.t the number of options will be considered too, increasing the number of options from 5 (the challenge setting) up to 50.

4.2 Experimental setting

The parameters used in the *Adapt-EvE* variants are listed in Table 2 together with their optimal values, which have been determined through systematic experiments in a predefined range. The runtime (on PC-Pentium 1.8Ghz) for 10^6 time steps, 5 options and 6 visitors is circa 40 seconds. While the parameter have been optimized for different kinds of dynamics (i.e. visitors) with several random seeds, the parameters are not locally but globally optimal. Small variations over their values has a little influence. The optimal PH setting does not depend on the other parameter values; the optimal

Role	Param.	Best value	Range
PH	δ	5.10^{-3}	$[10^{-3}, 10^{-1}]$
	λ	80	$[20, 120]$
Discount	ρ	$1 - 10^{-4}$	$1 - 10^{-i}, i = 2..7$
γ-Restart	γ	0	$[0, 50]$
Meta-Bandits	MT	1000	$[500, 1500]$

Table 2: Parameters of *Adapt-EvE*, ranges considered and optimal values. In addition to the parameters λ, δ and ρ in the PH test, the γ-restart strategy involves parameter γ (optimal value 0), the Meta-Bandit and the Meta-ρ-Bandit strategies involve parameter MT (optimal value 1000).

values of parameters δ (tolerance of the variation) and λ (false alarm rate) are constant in the range of the experiments. The optimal value of γ in the γ-restart strategy is 0; in other words, the best is to restart UCBT from scratch. With respect to Meta-Bandit and Meta-ρ-Bandit, the stopping criterion is given by the window time MT; while this parameter is fixed in the challenge setting, it becomes more sensitive when the number

of options is increased. Overall, the most sensitive parameter is the ρ discount factor, involved in the PH test, in the core UCBT, and in the Meta-ρ-Bandit.

5 Empirical validation

This section reports on the comparative performances of *Adapt-EvE* in the challenge setting and w.r.t. the number of options.

5.1 The EvE Challenge

The performance of the *Adapt-EvE* variants, combining the PH test with γ-restart, Meta-Bandit or Meta-ρ-Bandit, are reported together with the performance of UCBT and Discounted UCBT[3] (Kocsis & Szepesvari, 2006) in Table 3. The regrets over all visitors are displayed on Fig. 1 (top). All *Adapt-EvE* variants improve on UCBT and Discounted UCBT; the main cause of improvement seems to be the use of the PH-test. Comparatively, UCBT and even discounted UCBT are hindered by their inertia. The typical regret behavior of UCBT in dynamic environments is displayed on Fig. 1 and 2, bottom, considering the weekly close visitor. The regret periodically increases as the reward varies, followed by a plateau as UCBT catches up and updates the reward estimate.

Naturally, different types of variations are best handled by different algorithms, e.g. the constant visitor is best handled by UCBT.

Except for the constant visitor, Discounted UCBT significantly outperforms UCBT thanks to its lower inertia; still, the use of a fixed discount factor does not offer sufficient flexibility to cope with e.g. weekly and daily variations. The merits of using an explicit change-point detection test are visible as this allows the simple γ-Restart to significantly outperform the Discounted UCBT. The Meta-Bandit has an edge over γ-Restart when the variation schedule is irregular (in all cases except for the frequent swap and constant visitors); this is explained as the Meta-Bandit better recovers from false alarms.

Table 3 shows that the Meta-ρ-Bandit behaves much like the Meta-Bandit, except for the fact that it better handles the frequent swap visitor due to its discount factor. It is also clear that the regret experienced by a given algorithm strongly depends on the type of visitor; Fig. 3 displays the regret of the Meta-ρ-Bandit over all visitors and illustrates the difficulty of the frequent swap visitor.

5.2 Scalability w.r.t. the number of options

The robustness of *Adapt-EvE* is finally tested against a varying number of options. It is seen that increasing the number of options reinforces the differences of success between the different algorithms. The UCBT algorithm shows to perform more Exploitation, thus get the maximum reward possible when possible. The *Adapt-EvE* instead, tends to perform more exploration with the increasing number of options. The performance

[3]The optimal value for the discount factor ρ has been determined after the same experimental setting as for *Adapt-EvE*.

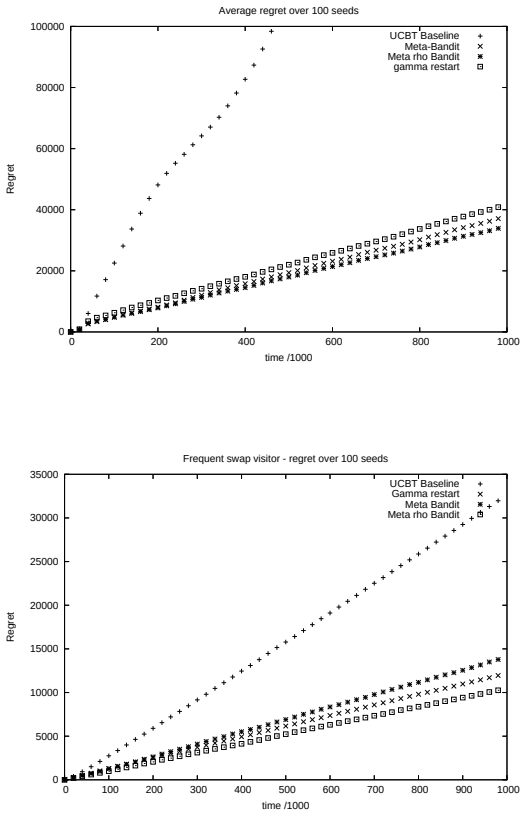

Figure 1: *Adapt-EvE*: Online regret on all visitors (top) and frequent swap (bottom), averaged on 100 runs.

of the different algorithms vary on the different kind of visitors met. We need to note that with the increasing number of options, we have an increasing number of changes in preferences, to a point where the cost of the restart outcast the exploration strength of the UCBT algorithm with certain visitor dynamics. The UCBT show better results with highly changing dynamics (frequent swap, daily variation) with several options and several changes, or more stationary cases (Constant visitor) while the *Adapt-EvE* proves best with other dynamics.

For instance, UCBT catches up and surpasses *Adapt-EvE* for the daily visitor when the number of options is above 30-40 (Fig. 4, top); in the case of the Long Gaussian visitor (Fig. 4, middle), the changes occur after a long interval of time and increasing the number of options does not allow UCBT to recover.

The regret of *Adapt-EvE* depending on the number of options, cumulated over all

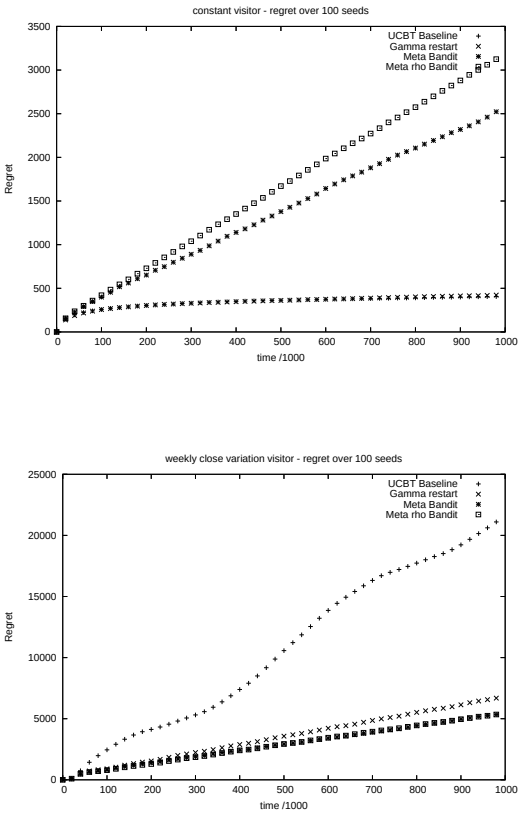

Figure 2: *Adapt-EvE*: Online regret on constant (top) and weekly close visitors (bottom), averaged on 100 runs.

types of visitors and averaged on 10 runs is reported in Table 4 and displayed on Fig. 4, bottom. In the considered context, UCBT catches up for the average visitor when the number of options is circa 50. The study also confirms the better robustness of Meta-ρ-Bandit compared to Meta-Bandit.

6 Conclusion and Perspectives

The contribution of this paper is twofold. On the one hand, it is suggested that the use of an external change-point detection test might be a simple and efficient way to deal with dynamic environments. On the other hand, the use of such tests raises new Exploration vs Exploitation dilemmas, about forgetting vs preserving the memory of the system.

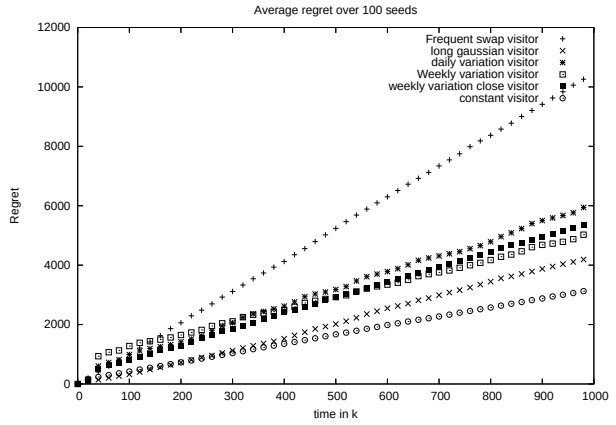

Figure 3: Meta-ρ-Bandit: Online regret w.r.t. all visitors (averaged over 100 runs).

	Baseline Algorithm	
	UCBT	Discount UCBT
Frequent Swap	32.6 ± 0.2	14.3 ± 0.1
Long	53.1 ± 4	7.6 ± 0.1
Daily Variation	60.2 ± 1.4	12.2 ± 0.1
Weekly Variation	62.2 ± 0.7	15 ± 0.2
Weekly Close Var.	21.6 ± 0.5	12 ± 0.2
Constant	0.4 ± 0.02	11.2 ± 0.04
Overall Regret	230 ± 4.5	72.5 ± 0.4

	γ-restart	Meta-Bandit	Meta-ρ-Bandit
Frequent Swap	12.1 ± 0.1	14.0 ± 1.9	10.6 ± 1.3
Long	7.4 ± 0.4	4.8 ± 1.6	4.3 ± 1.4
Daily Variation	6.9 ± 0.6	6.2 ± 0.7	6.1 ± 0.7
Weekly Variation	7.3 ± 0.2	4.8 ± 0.8	5.1 ± 0.9
Weekly Close Var.	6.6 ± 0.2	5.4 ± 0.8	5.5 ± 0.9
Constant	0.4 ± 0.02	2.5 ± 0.5	3.2 ± 0.3
Overall Regret	40.9 ± 0.8	37.7 ± 2.9	34.7 ± 2.3

Table 3: *Adapt-EvE*: Regret ($\times 10^{-3}$) for 10^6 time steps, considering 5 options and all visitors (averaged over 100 runs)

Interestingly, such EvE dilemmas can again be formalized and handled as multi-armed bandit problems.

Further work is concerned with the theoretical study of the PH test within the Meta-Bandit algorithm, providing bounds on the overall regret w.r.t. the dynamics of the environment. Another perspective is to investigate another type of variations in the environment, not considered in the present study, namely when another option becomes the best one though no change is seen on the current best option.

	UCBT	Meta-Bandit	Meta-ρ-Bandit
5 options	209.8	43.1	43.2
10 options	277.0	77.8	73.9
15 options	270.5	108.8	102.3
20 options	275.6	136.0	124.0
25 options	258.9	157.7	141.9
30 options	249.8	171.7	155.4
35 options	248.0	187.5	169.9
40 options	229.6	200.4	182.9
45 options	222.7	210.2	192.4
50 options	219.5	222.1	199.5

Table 4: *Adapt-EvE*: Regret ($\times 10^{-3}$) over 10^6 time steps for 5 to 50 options (average over 10 runs).

Acknowledgment

This work was supported in part by the PASCAL Network of Excellence.

References

AUER P., CESA-BIANCHI N. & FISCHER P. (2002). Finite-time analysis of the multi-armed bandit problem. *Machine Learning*, **47**(2/3), 235–256.

AUER P., CESA-BIANCHI N., FREUND Y. & SCHAPIRE R. E. (1995). Gambling in a rigged casino: the adversarial multi-armed bandit problem. In *Proceedings of the 36th Annual Symposium on Foundations of Computer Science*, p. 322–331: IEEE Computer Society Press, Los Alamitos, CA.

AUER P. & ORTNER R. (2007). Logarithmic online regret bounds for undiscounted reinforcement learning. In B. SCHÖLKOPF, J. PLATT & T. HOFFMAN, Eds., *Advances in Neural Information Processing Systems 19*. Cambridge, MA: MIT Press.

BASSEVILLE M. (1988). Detecting changes in signals and systems - a survey. *Automatica*, **24**, 309–326.

CESA-BIANCHI N. & LUGOSI G. (2006). *Prediction, learning, and games*. Cambridge University Press.

CORMODE G. & MUTHUKRISHNAN S. (2005). An improved data stream summary: the count-min sketch and its applications. *J. Algorithms*, **55**(1), 58–75.

DRAGALIN V., TARTAKOVSKY A. & VEERAVALLI V. (1999 (part I)/2000 (part II)). Multihypothesis sequential probability ratio tests: accurate asymptotic expansions for the expected sample size.

GRCAR M., MLADENIC D. & GROBELNIK M. (2005). User profiling for interest-focused browsing history. In *Proceedings of UserSWeb05*.

HADJILIADIS O. & MOUSTAKIDES G. (2006). Optimal and asymptotically optimal cusum rules for change point detection in the brownian motion model with multiple alternatives. *Theory of Probability and its Applications*, **50**(1), 131–144.

HINKLEY D. (1969). Inference about the change point in a sequence of random variables. *Biometrika*, **57**(1), 1–17.

HINKLEY D. (1970). Inference about the change point from cumulative sum-tests. *Biometrika*, **58**(3), 509–523.

HINKLEY D. (1971). Inference in two-phase regression. *Journal of the American Statistical Association*, **66**(336), 736–743.

HUSSAIN Z., AUER P., CESA-BIANCHI N, NEWNHAM L. & SHAWE-TAYLOR J. (2006). Exploration vs. exploitation challenge. In *http://www.pascal-network.org/Challenges/EEC/*

KIFER D., BEN-DAVID S. & GEHRKE J. (2004). Detecting change in data streams. In *Proc. VLDB'04*, p. 180–191: Morgan Kaufmann.

KOCSIS L. & SZEPESVARI C. (2005). Reduced-variance payoff estimation in adversarial bandit problems. In *Proceedings of the ECML-2005 Workshop on Reinforcement Learning in Non-Stationary Environments*.

KOCSIS L. & SZEPESVARI C. (2006). Discounted-UCB. In *2nd Pascal-Challenge Workshop, Venice*.

LORDEN G. (1971). Procedures for reacting to a change in distribution. *Annals of Mathematical Statistics*, **42**, 1897–1908.

MOUSS H., MOUSS D., MOUSS N. & SEFOUHI L. (2004). Test of Page-Hinkley, an approach for fault detection in an agro-alimentary production system. In *5th Asian Control Conference*, p. 815– 818.

MOUSTAKIDES G. (1986). Optimal stopping times for detecting changes in distributions. *Anals of Statistics*, **14**, 1379–1387.

PAGE E. (1954). Continuous inspection schemes. *Biometrika*, **41**, 100–115.

PIRIOU G., COLDEFY F., BOUTHEMY P. & YAO J.-F. (2004). Détection supervisée d'événements à l'aide d'une modélisation probabiliste du mouvement perçu. In *14e Congrès Francophone AFRIF-AFIA de Reconnaissance des Formes et Intelligence Artificielle, RFIA 2004*, Toulouse, France.

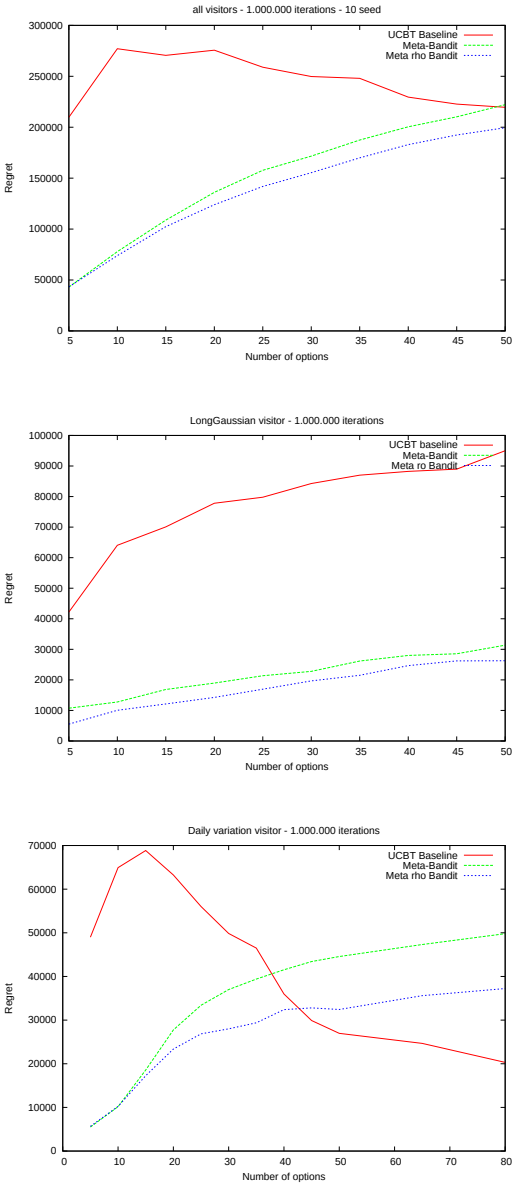

Figure 4: *Adapt-EvE*: Regret for 10^6 time steps vs the number of options (All visitors, Gaussian and Daily Visitors, top to bottom; average on 10 runs).